IN THE CLAIMS

Claim Listing

1-36. (Canceled)

37. (Previously presented) A method in a data processing system comprising:

examining the nodes in a plurality of hierarchical trees;

determining if a node is present in only one tree by comparing two or more of the hierarchical trees; and

creating a merged tree based on the nodes in the hierarchical trees.

38. (Previously presented) The method of claim 37 further comprising:

creating a reference node to the node determined to be present in only one tree if a node is determined to be present in only one tree; and

adding the reference node to the merged tree.

39. (Previously presented) The method of claim 38 further comprising:

creating the reference node in response to a determination that a node is present in only one tree.

40. (Previously presented) The method of claim 38 wherein the reference node is a pointer.

41. (Previously presented) The method of claim 38 wherein the reference node is a Java reference.

42. (Previously presented) The method of claim 37 further comprising:

determining if the hierarchical trees comprise a set of equivalent nodes.

43. (Previously presented) The method of claim 42 further comprising:

selecting the node with the highest priority from the set of equivalent nodes if the hierarchical trees comprise a set of equivalent nodes.

44. (Previously presented) The method of claim 43 further comprising:

creating a shallow clone of the selected node; and

adding the shallow clone to the merged tree.

45. (Previously presented) The method of claim 37 wherein the hierarchical trees comprise a group tree and a user tree.

46. (Previously presented) The method of claim 37 wherein the hierarchical trees comprise a group tree and an admin tree.

47. (Previously presented) The method of claim 37 wherein the hierarchical trees comprise a user tree and an admin tree.

48. (Previously presented) The method of claim 37 wherein the hierarchical trees comprise a group tree, a user tree, and an admin tree.

49. (Previously presented) The method of claim 37 wherein the hierarchical trees are DOM trees.

50. (Previously presented) The method of claim 49 wherein the DOM trees are XML DOM trees.

51. (Previously presented) The method of claim 37 further comprising:

printing the merged tree.

52. (Previously presented) A data processing system comprising:

a memory comprising a program that:

examines the nodes in a plurality of hierarchical trees;

determines if a node is present in only one tree by comparing two or more of the hierarchical trees; and

creates a merged tree based on the nodes in the hierarchical trees; and

a processor for running the program.

53. (Previously presented) The data processing system of claim 52 wherein the program further:

creates a reference node to the node determined to be present in only one tree if a node is determined to be present in only one tree; and

adds the reference node to the merged tree.

54. (Previously presented) The data processing system of claim 53 wherein the program further:

creates the reference node in response to a determination that a node is present in only one tree.

55. (Previously presented) The data processing system of claim 53 wherein the reference node is a pointer.

56. (Previously presented) The data processing system of claim 53 wherein the reference node is a Java reference.

57. (Previously presented) The data processing system of claim 52 wherein the program further:

determines if the hierarchical trees comprise a set of equivalent nodes.

58. (Previously presented) The data processing system of claim 57 wherein the program further:

selects the node with the highest priority from the set of equivalent nodes if the hierarchical trees comprise a set of equivalent nodes.

59. (Previously presented) The data processing system of claim 58 wherein the program further:

creates a shallow clone of the selected node; and

adds the shallow clone to the merged tree.

60. (Previously presented) The data processing system of claim 52 wherein the hierarchical trees comprise a group tree and a user tree.

61. (Previously presented) The data processing system of claim 52 wherein the hierarchical trees comprise a group tree and an admin tree.

62. (Previously presented) The data processing system of claim 52 wherein the hierarchical trees comprise a user tree and an admin tree.

63. (Previously presented) The data processing system of claim 52 wherein the hierarchical trees comprise a group tree, a user tree, and an admin tree.

64. (Previously presented) The data processing system of claim 52 wherein the hierarchical trees are DOM trees.

65. (Previously presented) The data processing system of claim 64 wherein the DOM trees are XML DOM trees.

66. (Previously presented) The data processing system of claim 52 wherein the program further:

prints the merged tree.

67. (Previously presented) A computer-readable medium comprising instructions for controlling a data processing system to perform a method comprising the steps of:

examining the nodes in a plurality of hierarchical trees;

determining if a node is present in only one tree by comparing two or more of the hierarchical trees; and

creating a merged tree based on the nodes in the hierarchical trees.

68. (Previously presented) The computer-readable medium of claim 67 wherein the method further comprises the steps of:

creating a reference node to the node determined to be present in only one tree if a node is determined to be present in only one tree; and

adding the reference node to the merged tree.

69. (Previously presented) The computer-readable medium of claim 68 wherein the method further comprises the step of:

creating the reference node in response to a determination that a node is present in only one tree.

70. (Previously presented) The computer-readable medium of claim 68 wherein the reference node is a pointer.

71. (Previously presented) The computer-readable medium of claim 68 wherein the reference node is a Java reference.

72. (Previously presented) The computer-readable medium of claim 67 wherein the method further comprises the step of:

determining if the hierarchical trees comprise a set of equivalent nodes.

73. (Previously presented) The computer-readable medium of claim 72 wherein the method further comprises the step of:

selecting the node with the highest priority from the set of equivalent nodes if the hierarchical trees comprise a set of equivalent nodes.

74. (Previously presented) The computer-readable medium of claim 73 wherein the method further comprises the steps of:

creating a shallow clone of the selected node; and

adding the shallow clone to the merged tree.

75. (Previously presented) The computer-readable medium of claim 67 wherein the hierarchical trees comprise a group tree and a user tree.

76. (Previously presented) The computer-readable medium of claim 67 wherein the hierarchical trees comprise a group tree and an admin tree.

77. (Previously presented) The computer-readable medium of claim 67 wherein the hierarchical trees comprise a user tree and an admin tree.

78. (Previously presented) The computer-readable medium of claim 67 wherein the hierarchical trees comprise a group tree, a user tree, and an admin tree.

79. (Previously presented) The computer-readable medium of claim 67 wherein the hierarchical trees are DOM trees.

80. (Previously presented) The computer-readable medium of claim 79 wherein the DOM trees are XML DOM trees.

81. (Previously presented) The computer-readable medium of claim 67 wherein the method further comprises the step of:

printing the merged tree.